Use Cases

for

extraVEGANza

Version 1.0 approved

Prepared by Li Lin, Bryan Lu

Team Anyhow Anything Anywhere

28th January 2023

Revision History

Name	Date	Reason For Changes	Version

1. Select Dietary Restrictions

Use Case ID:	1.1		
Use Case Name:	Select Dietary Restrictions		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	28/1/2023	Date Last Updated:	12/2/2023

Actor:	User (Primary), Application (Secondary)
Description:	This use case allows the user to select their own dietary restrictions to filter the list of restaurants that they are searching or community posts.
Preconditions:	1. The application must be opened by the user.
Postconditions:	1. The user must already have selected one or more dietary restrictions.
Priority:	1
Frequency of Use:	1 to 2 times per use of the application
Flow of Events:	1. The application provides the user with dietary restrictions options. 2. The user selects and deselects the dietary restrictions that user wants by clicking on the checkbox. 3. The application confirms the updated dietary restrictions by displaying a “Updated” message to the user. 4. The application updates the relevant content of the page accordingly.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A

Assumptions:	1. If none of the dietary restrictions options are chosen, the application assumes that the user has selected all the options. 2. If the application has just been loaded, all of the options are initially unchecked.
Notes and Issues:	

2. List Shortlisted Restaurant

Use Case ID:	1.2		
Use Case Name:	List Shortlisted Restaurant		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	28/1/2023	Date Last Updated:	12/2/2023

Actor:	Restaurant Database (Primary), Application (Secondary)
Description:	After the user has selected their dietary restrictions, the application will search for restaurants that correspond with the dietary restrictions.
Preconditions:	1. The user must already have selected their dietary restrictions. OR 2. The webpage has just been loaded.
Postconditions:	1. A list of restaurants must be returned to the application for display. OR 2. A message must be sent back to the application if it is unable to connect to the database.
Priority:	2
Frequency of Use:	1 to 3 times per use of the application

Flow of Events:	1. The application sends a query to the Restaurant Database using its API for relevant results. 2. The Restaurant Database finds all restaurants that fit the dietary restriction requirements. 3. The Restaurant Database sends a list of all relevant restaurants back to the application. 4. The application receives the data. 5. The application proceeds with displaying the data in its use case.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

3. Choose Sort Order

Use Case ID:	1.3		
Use Case Name:	Choose Sort Order		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	7/2/2023	Date Last Updated:	12/2/2023

Actor:	User (Prianry), Application (Secondary)
Description:	This use case allows the user to view the list of restaurants in three orders: by alphabetical ascending order (A - Z), by alphabetical

	descending order (Z - A), and by ratings of the restaurant. This can be done by clicking on the sorting buttons displayed on the UI.
Preconditions:	1. The application must only be in the restaurant list view. 2. The application must already be displaying the list of restaurants.
Postconditions:	1. The order of the list of restaurants must be changed based on the criteria selected, if and only if the sorting order has been altered.
Priority:	1
Frequency of Use:	1 to 2 times per use of the application
Flow of Events:	1. The user clicks on the sorting icon. 2. The application provides the user with a dropdown menu with three choices of sorting: by alphabetical ascending order (A - Z), by alphabetical descending order (Z - A), and by ratings of the restaurant. 3. The user selects one of the choices to sort the list of restaurants. 4. The application reorders the list of restaurants based on sorting order.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

4. Search Restaurant

Use Case ID:	1.4

Use Case Name:	Search Restaurant		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	7/2/2023	Date Last Updated:	12/2/2023

Actor:	User (Primary), Application (Secondary), Restaurant Database (Secondary)
Description:	This use case collects restaurant keywords from the user, and uses it to search for restaurants relevant to the keywords.
Preconditions:	1. The user must be in the restaurant map view or restaurant list view.
Postconditions:	1. A list of restaurants must be returned to the application for display. OR 2. A message must be sent back to the application if it is unable to connect to the database.
Priority:	2
Frequency of Use:	1 to 4 times per use of application
Flow of Events:	1. The user enters their interested keywords into the text box. 2. The user confirms the search keywords by hitting the “Enter” key, or by clicking on the search icon. 3. The application sends a query to the Restaurant Database using its API for relevant results. 4. The Restaurant Database finds all restaurants that are relevant to the keywords. 5. The Restaurant Database sends a list of all relevant restaurants back to the application. 6. The application receives the data. 7. The application proceeds with displaying the information.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A

Assumptions:	N/A
Notes and Issues:	

5. Select Preferred Restaurant

Use Case ID:	1.5		
Use Case Name:	Select Preferred Restaurant		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	7/2/2023	Date Last Updated:	12/2/2023

Actor:	User (Primary), Application (Secondary)
Description:	This use case allows the user to select their preferred restaurant from the application UI.
Preconditions:	1. The user must be in the restaurant map view, restaurant list view, or community post view.
Postconditions:	1. The user must have selected a restaurant for a more detailed information view.
Priority:	1
Frequency of Use:	1 to 4 times per use of application
Flow of Events:	1. The user clicks on the restaurant displayed on the application. 2. The application displays a brief summary popup of the restaurant. 3. The user clicks on the “More info” button to look for more detailed information about the restaurant. 4. The application proceeds to display the information of the restaurant in the Show Restaurant Information use case.

Alternative Flows:	AF5-S3: If the user clicks on anywhere else on the screen apart from the popup. 1. The application closes the popup. 2. Go to Step 1.
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

6. Show Restaurant Information

Use Case ID:	1.6		
Use Case Name:	Show Restaurant Information		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	7/2/2023	Date Last Updated:	12/2/2023

Actor:	Restaurant Database (Primary), Application (Secondary)
Description:	This use case allows the application to display the restaurant information to the user.
Preconditions:	1. The user must have already selected their preferred restaurant.
Postconditions:	1. The information of the restaurant must be returned to the application and be displayed. OR 2. A message must be sent back to the application if it is unable to connect to the database.

Priority:	2
Frequency of Use:	1 to 4 times per use of application
Flow of Events:	1. The application sends a query to the Restaurant Database using its API for the full information of the restaurant. 2. The Restaurant Database sends the information of the requested restaurant back to the application. 3. The application receives the data. 4. The application displays the information of the restaurant.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

7. Show Restaurant Menu

Use Case ID:	1.7		
Use Case Name:	Show Restaurant Menu		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	7/2/2023	Date Last Updated:	12/2/2023

Actor:	Application (Primary)
Description:	This use case allows the application to display the restaurant menu to the user.
Preconditions:	1. The user must have already opened up the restaurant information page.

Postconditions:	1. The information of the restaurant must be returned to the application and be displayed. OR 2. A message must be sent back to the application if it is unable to connect to the database.
Priority:	1
Frequency of Use:	1 to 4 times per use of application
Flow of Events:	1. The application sends a query to the Restaurant Database using its API for the menu of the restaurant. 2. The Restaurant Database sends the information of the requested restaurant back to the application. 3. The application receives the data. 4. The application displays the menu of the restaurant.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

8. Navigate to Restaurant

Use Case ID:	1.8		
Use Case Name:	Navigate to Restaurant		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	7/2/2023	Date Last Updated:	12/2/2023

Actor:	Google Maps (Primary), Web Application (Secondary), User (Secondary)
Description:	After the user selects a restaurant and chooses to navigate to it on the application, the Google Map application will help the user to reach the selected restaurant.
Preconditions:	1. A restaurant must be selected and chosen to be navigated to by the user. 2. The application must send the user's current location data to Google Map.
Postconditions:	1. The user must reach their destination by following Google Maps navigation instructions. OR 2. The user has canceled the navigation.
Priority:	1
Frequency of Use:	1 time per use of application
Flow of Events:	1. The application sends the selected restaurant address to Google Maps. 2. An embedded frame of Google Maps appears in the application's UI with the restaurant address set as destination. 3. The user clicks "Navigate" to start navigation mode. 4. The user follows the direction based on Google Maps to reach the destination.
Alternative Flows:	AF8-S3: Users can share navigation details to their phone. 1. The user clicks on the "Send directions to your phone" to receive navigation instructions on the user's phone for further navigation. 2. Go to Step 4.
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

9. Access Community Page

Use Case ID:	Access Community Page		
Use Case Name:	1.9		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	7/2/2023	Date Last Updated:	12/2/2023

Actor:	User (Primary), Application (Secondary), Database (Secondary)
Description:	This use case allows the user to access the community page of the application.
Preconditions:	1. The application must be opened by the user.
Postconditions:	1. The community posts must be returned to the application and be displayed. OR 2. A message must be sent back to the application if it is unable to connect to the database.
Priority:	2
Frequency of Use:	1 to 3 times per use of application
Flow of Events:	1. The user clicks on the community page button. 2. The application sends a query to the Database using its API for the community posts. 3. The Database sends the information of the community posts back to the application. 4. The application receives the data. 5. The application displays the community posts.
Alternative Flows:	N/A
Exceptions:	N/A

Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

10. Share Recommendation

Use Case ID:	1.10		
Use Case Name:	Share Recommendation		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	7/2/2023	Date Last Updated:	12/2/2023

Actor:	User (Primary), Application (Secondary), Database (Secondary)
Description:	This use case allows the user to post a recommendation on the community page.
Preconditions:	1. The user must have already logged in to their account. 2. The user must open the community page.
Postconditions:	1. The application must successfully record the user's post. OR 2. A message must be sent back to the application if it is unable to connect to the database.
Priority:	2
Frequency of Use:	1 time per use of application
Flow of Events:	1. The user clicks on the "share" button. 2. The application displays a post submission form popup to the user.

	3. The application requests the user to complete the form. 4. The user completes the form and submits it. 5. The application submits the form to the database. 6. The database retrieves the information from the application, stores the information, and returns a success notification to the application. 7. The application acknowledges the success notification and displays a success message to the user. 8. The application returns to the community page..
Alternative Flows:	AF10-S4: If the user submits an incomplete form 1. The application requests the user to complete the form. 2. Go to Step 4. AF10.1: If the user wanted to cancel submission attempt 1. The user clicks on the return button. 2. Go to Step 8. AF10-S5: If connection to the database cannot be achieved 1. The application displays a “connection error” message. 2. The application awaits for a stable connection to the database. 3. Repeat Step 5. AF10-S7: If the application receives a failed submission notification from the database. 1. The application requests the user to try again later. 2. Go to Step 3.
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

11. Log In

Use Case ID:	1.11		
Use Case Name:	Log In		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	7/2/2023	Date Last Updated:	12/2/2023

Actor:	User (Primary), Application (Secondary), Database (Secondary)
Description:	This use case allows the user to log in to their account.
Preconditions:	1. The user must open the account log in page. 2. The user must have not already logged in to any account, otherwise the login page will not be shown.
Postconditions:	1. The application has successfully logged in the user to their account.
Priority:	3
Frequency of Use:	1 time per use of application
Flow of Events:	1. The application displays text boxes and requests the user to input their username and their password. 2. The user enters their username and password. 3. The application sends a query to the database using its API. 4. The database validates the username and password. 5. The database returns a successful notification to the application if the username and password are valid. 6. The application logs in the user to their account. 7. The application displays the account settings to the user.
Alternative Flows:	AF11-S5: If the user enters an invalid username or password 1. The database returns an unsuccessful notification to the application if the username and password are invalid. 2. The application requests the user to input their username and their password. 3. Go to Step 2. AF11.1: If the user wants to return back 1. The webpage displays an invalid username and password message to the user. 2. The user clicks on the return button, or other buttons, to exit the login screen.

	AF11-S3: If the user forgets their password 1. The user clicks on the forgot password button. 2. The user receives an email to reset their password. 3. Go to Step 3.
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

12. Access Settings

Use Case ID:	1.12		
Use Case Name:	Access Settings		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	7/2/2023	Date Last Updated:	12/2/2023

Actor:	User (Primary), Application (Secondary), Database (Secondary)
Description:	This use case allows the user to access the settings of the web application.
Preconditions:	1. The user must have already opened the application.
Postconditions:	1. The application has updated its settings. OR 2. A message must be sent back to the application if it is unable to connect to the database.
Priority:	2

Frequency of Use:	1 to 2 times per use of application
Flow of Events:	1. The user clicks on the settings button. 2. The application shows the default settings to the user; if the user has logged in to their account, the user settings would be shown instead. 3. The user changes the settings. 4. The user clicks the “OK” button to confirm settings changes. 5. The application sends the updated settings to the database. 6. The database confirms the settings change with the application. 7. The application updates accordingly to the settings. 8. The application displays the previous page to the user.
Alternative Flows:	AF12.1: If altered settings cannot be updated on the database end 1. The application repeatedly attempts to communicate with the database in the background to update the settings, until the settings change in the database is successful, or until the application is closed, of which the database would not update the settings changed.
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

13. Access Help

Use Case ID:	1.13		
Use Case Name:	Access Help		
Created By:	Bryan Lu	Last Updated By:	Bryan Lu

Date Created:	7/2/2023	Date Last Updated:	12/2/2023

Actor:	User (Primary), Application (Secondary)
Description:	This use case allows the user to access the help page of the application.
Preconditions:	1. The user must have already opened the application.
Postconditions:	1. The user must close the help page.
Priority:	1
Frequency of Use:	1 to 2 times per use of application
Flow of Events:	1. The user clicks on the help button. 2. The application shows the help page to the user. 3. The user reads the help information on the page. 4. The user clicks on the return button to return to the previous page.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

14. Access Account

Use Case ID:	1.14
Use Case Name:	Access Account

Created By:	Bryan Lu	Last Updated By:	Bryan Lu
Date Created:	7/2/2023	Date Last Updated:	12/2/2023

Actor:	User (Primary), Application (Secondary), Database (Secondary)
Description:	This use case allows the user to manage their account.
Preconditions:	1. The user must have already logged in to their account.
Postconditions:	1. The application must successfully record the user's post. OR 2. A message must be sent back to the application if it is unable to connect to the database.
Priority:	2
Frequency of Use:	1 to 2 times per use of application
Flow of Events:	1. The user clicks on the account button. 2. The application displays the account management dashboard to the user. 3. The user changes the settings on the dashboard. 4. The user clicks the "OK" button to confirm changes. 5. The application sends the updated settings to the database. 6. The database confirms the settings change with the application. 7. The application displays the previous page to the user.
Alternative Flows:	AF14.1: If altered settings cannot be updated on the database end 2. The application repeatedly attempts to communicate with the database in the background to update the settings, until the settings change in the database is successful, or until the application is closed, of which the database would not update the settings changed.
Exceptions:	N/A
Includes:	N/A
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

Use Case Template

Use Case ID:			
Use Case Name:			
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	
Description:	
Preconditions:	
Postconditions:	
Priority:	
Frequency of Use:	
Flow of Events:	
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case Template

Use Case ID:			
Use Case Name:			
Created By:		Last Updated By:	
Date Created:		Date Last Updated:	

Actor:	
Description:	
Preconditions:	
Postconditions:	
Priority:	
Frequency of Use:	
Flow of Events:	
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	